

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-119457
 (43)Date of publication of application : 20.04.1992

(51)Int.Cl. G06F 15/20

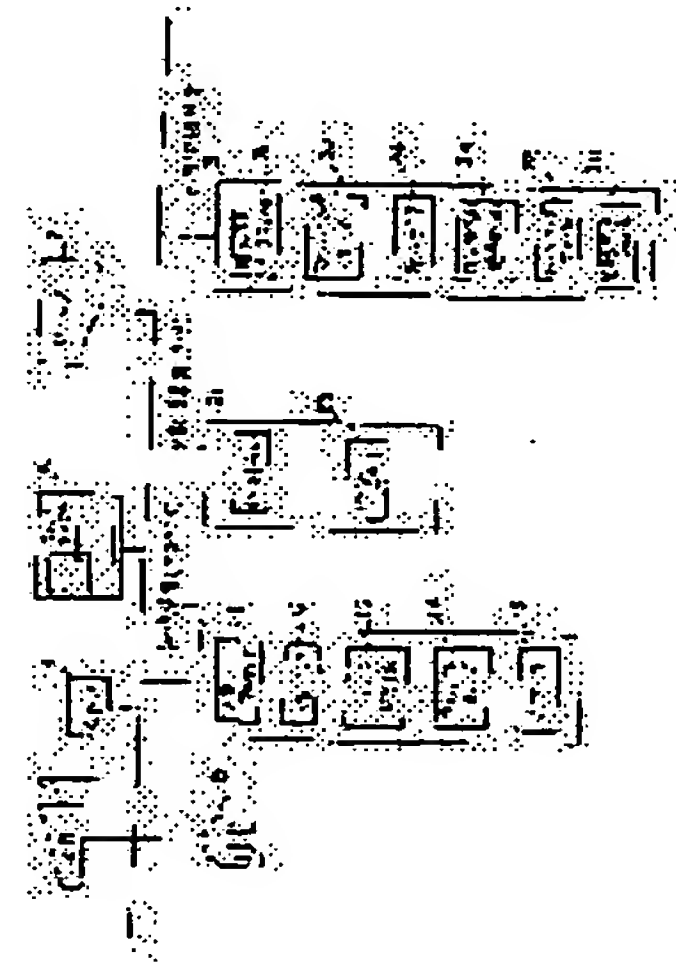
(21)Application number : 02-239052 (71)Applicant : CANON INC
 (22)Date of filing : 11.09.1990 (72)Inventor : SHIMIZU HARUO
 YOSHIZAKI OSAMU
 OGA MANABU

(54) DOCUMENT PROCESSOR

(57)Abstract:

PURPOSE: To edit a color text, a color picture and a color image as well as to output the result of these editing corresponding to the functions of a display device and a printer to be connected by providing a means outputting color documents including a typesetted graphic, text and picture corresponding to the performance of the display device and the printing device.

CONSTITUTION: A instruction means 9 instructs how to shape and typeset the color text, color graphic and color picture in a situation independing from a display device 6 and a printer 7, and a processing means 4 shapes and typesets a document including the text, graphic and image based on the contents of an instruction by the instruction means 9. An output control means 4 outputs the color document including the typesetted graphic, text and picture corresponding to the performance of the display device 6 or the printer 7. That is, when an output is executed by means of the display device 6 or the printer 7, the output control means 4 controls the output to be correspondent to the performance of the display device 6 or the printer 7. Thus, an environment in which the graphic and picture are colored and integrated can be obtained.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

平4-119457

⑬ Int. Cl.⁵

識別記号

庁内整理番号

⑭ 公開 平成4年(1992)4月20日

G 06 F 15/20

5 4 6 A

6914-5L

審査請求 未請求 請求項の数 1 (全 14 頁)

⑮ 発明の名称 文書処理装置

⑯ 特 願 平2-239052

⑰ 出 願 平2(1990)9月11日

⑱ 発 明 者	清 水	治 夫	東京都大田区下丸子3丁目30番2号	キヤノン株式会社内
⑱ 発 明 者	吉 崎	修	東京都大田区下丸子3丁目30番2号	キヤノン株式会社内
⑱ 発 明 者	大 賀	孝	東京都大田区下丸子3丁目30番2号	キヤノン株式会社内
⑲ 出 願 人	キヤノン株式会社		東京都大田区下丸子3丁目30番2号	
⑳ 代 理 人	弁理士 大塚 康徳		外1名	

明 細 書

1. 発明の名称

文書処理装置

2. 特許請求の範囲

表示装置及び印刷装置を接続し、カラーテキストとカラー図形とカラー画像を整形及び組版した文書を作成する文書処理装置であつて、

カラーテキストとカラー図形とカラー画像をどのように整形・組み版するかを前記表示装置や前記印刷装置に依存しない状態で指示する指示手段と、

該指示手段の指示内容に基づき、テキスト、図形、イメージを含む文書を整形・組み版する処理手段と、

この組み版された図形、テキスト、画像を含むカラー文書を前記表示装置やあるいは前記印刷装置の性能に合わせて出力する出力手段とを備えることを特徴とする文書処理装置。

3. 発明の詳細な説明

[産業上の利用分野]

本発明は文書の作成・編集を行なう文書処理装置に関するものである。

[従来の技術]

近年のこの種の装置、特にDTP分野においては、文字のページ内での最適配置を計算する処理、すなわち、組み版処理の形態には二種類ある。

1つはWYSIWYG(What You See Is What You Get)システムであり、広く普及しているワードプロセッサがこれに対応する。

このシステムは、文書作成者が最終印刷形態を画面上で見ながら対応する位置に対応する文字サイズで文字入力するシステムである。このシステムにおける長所及び短所としては、以下のような点が挙げられる。

長所:

- ・ユーザが最終画面を見ながら文字入力するのでほぼ自分の思い通りの文書作成が可能。

- ・ユーザインターフェースに優れている。
- ・短い文書作成が容易。

短所：

- ・大部の文書での統一的な取り扱いに難（章を一定の文字サイズで書く e t c）。
- ・目次や索引作成にはバッチ的処理が必要。
- ・他システムでの文書再利用が困難。
- ・専用装置が必要。

もう1つはバッチ型システムでスタンフォード大学の K u r t h 教授によつて作成された T E X や、A T / T 社で開発された r o f i システムがこれに対応する（文献：木村泉：“文書整形言語”情報処理 P.559 ～P.654, Jul, 1981）。第2図を用いてその過程を説明する。これらのシステムは文書を組み版する指示を含むソースファイル 5 1 をテキストエディタ 5 0 で作成して、それをコンパイル 5 2 して、組み版した中間ファイル 5 3 を作成し、その結果を画面上で表示 5 4 したり、プリンタで印刷 5 5 したりする。

このシステムの特徴として、

3

れている。一方、バッチ型のシステムでは、文書をプログラムする事を直接ユーザに実行させる代わりに、構造化エディタをかぶせてユーザに次のようなコマンドを入力できるかをアイコン等で示して、組み版言語に精通している必要をなくしている。

このようにして最近の文書組み版処理は W Y S I W Y G とバッチ型のハイブリッドな様相を呈している。

【発明が解決しようとする課題】

しかしながら、これらバッチ、W Y S I W Y G 型の両方のシステムでは、スライド、O H P e t c の特殊用途を除いて、文書自体は白黒で作成される事が多かった。

特に、バッチ処理型のシステムにおいては、カラー図形やカラー画像を装置独立な形式で文書に取り入れる事が従来から困難であつた。

しかしながら、一方では、最近の周辺装置やモニタのカラー化等による高機能化により、テキストや図形、イメージの混在したカラー文書を取り

長所：

- ・大部の文書で統一的な取扱いが容易。
- ・目次や索引作成の機能を取り込みやすい。
- ・通常のテキスト端末を持つワークステーション、パーソナルコンピュータでも動作可能。

短所：

- ・文書をプログラムするような型なので、素人には難しい。
- ・処理結果としては、それをコンパイルし表示しないとわからない。
- ・短い文章だと W Y S I W Y G で作成した方が簡単。

この比較で見ると、W Y S I W Y G やバッチ型の文書処理システムはそれぞれの長所と短所があるが、近年では、それぞれ自分の欠点を解消しようとシステムの拡張を行っている。例えば W Y S I W Y G 型のシステムに目次や索引の一括処理機能を導入したり、スタイルシートを導入して全ページに渡つて同じページスタイルで文書を作成するといったバッチ型システムの長所を取り入

4

扱える、すなわち印刷と組み版結果のプレ表示の必要性が高く望まれている。

ここで、カラー図形やカラーイメージ及びカラーテキストを装置固有の方法で利用出来ても、他システム間とのデータ交換性は保証されないという欠点が存在する。

本発明はかかる従来技術に鑑みなされたものであり、カラーテキスト、カラー画像更にはカラーイメージを編集することができると共に、それらの編集結果を接続される表示装置や印刷装置の機能に合わせて出力することを可能ならしめる文書処理装置を提供しようとするものである。

【課題を解決するための手段】

この課題を解決するため本発明の文書処理装置は以下に示す構成を備える。すなわち、

表示装置及び印刷装置を接続し、カラーテキストとカラー図形とカラー画像を整形及び組版した文書を作成する文書処理装置であつて、カラーテキストとカラー図形とカラー画像をどのように整形・組版するかを前記表示装置や前記印刷装置

に依存しない状態で指示する指示手段と、該指示手段の指示内容に基づき、テキスト、図形、イメージを含む文書を整形・組み版する処理手段と、この組み版された図形、テキスト、画像を含むカラー文書を前記表示装置やあるいは前記印刷装置の性能に合わせて出力する出力制御手段とを備える。

〔作用〕

かかる本発明の構成において、指示手段の指示内容に基づいて処理手段がテキスト、図形、イメージを含む文章の整形・組版を行う。そして、それを表示装置や印刷装置に出力するときには、出力制御手段が前記表示装置やあるいは前記印刷装置の性能に合わせて出力するよう制御する。

〔実施例〕

以下、添付図面に従って本発明に係る実施例を詳細に説明する。

第1図は本実施例における文書処理システムのブロック構成図である。

図中、1は本システムの処理手順(第7、10

7

イメージデータファイル35及び図形データファイル36を入力として組み版結果をマルチウインドウシステム6に表示するプレビューである。13は組み版結果をプリンタ7に出力するための印刷ドライバ(印刷専用のプログラム)である。14はウインドウシステム6を動作させるためのウインドウサーバプログラム、15は印刷の命令が発行された場合、その起動順序に従って印刷を行うスプーラである。

記憶装置2において、21はウインドウシステムでのユーザからのマウス8やキーボード9を利用した要求をキューイングするイベントキューである。プリンタ(記録装置)7で記録する際、その装置が例えばPost Scriptなどのページ記述言語を装備していないと、ホスト側でその1ページのイメージを作成する必要がある。22はそのための設けられたイメージ展開用のページメモリであり、記録対象の画像(文字や図形等)はその上に直接描画する。

外部記憶装置3において、31は組み版を行う

図のプログラム等)を記憶するための記憶装置、2は本システムの処理に必要な情報を記憶するための記憶装置、3は本システムの入力または出力データを記憶するための外部記憶装置、4は記憶装置1に記憶されている処理手順に従って処理を行うためのCPUである。6は本システムの処理結果を表示するためのマルチウインドウシステム、7は本システムの処理結果を印刷するためのプリンタ、8はユーザからの指令を入力するマウス、そして9はユーザがプログラムを作成したり、本システムに指令を入力するためのキーボードである。

記憶装置1において、11は文書フォーマッタ(組み版プログラム)であり、後述する組み版ソースファイル31からのデータ、フォントメトリックデータ32、イメージデータ35の利用サイズ、そして図形データ36の利用サイズを入力し、組み版結果ファイル(DVI)を出力する。12は文書フォーマッタ11によつて作成されたDVIファイル34とフォントデータ33、イ

8

際に、命令+文字情報+図形やイメージ配置情報を含んでいる文書組み版ソースファイルである。32は文字組み版を行う際に必要な文字のメトリック情報(文字高さ、文字深さ、文字幅)を含むファイルである。33は文字を印字する際に必要となる文字のドットパターン又はアウトライン情報を記憶しているフォントデータである。34は文書フォーマッタ11の組み版処理の結果を格納するディスプレイやプリンタの解像度etcに依存しないDVI(Device Independent)ファイルである。35は文書に付加されるカラー画像情報を含むイメージデータファイルであり、36は同様にカラー図形(線画等)を含む図形データファイルである。

上述した構成よりなる本実施例システムのメインフローを第3図を参照しながら簡単に説明する。

このフローは基本的に第2図の従来のバッチ型テキスト組み版システムに類似しているが、新規な処理として以下のものを備えた。

(1) 組み版処理のソースファイル 9 1 に、カラー図形 9 2 やカラー画像 9 3 情報をどのように取り込むか(どこの位置にどの大きさ)の指定を出来るようにした。

(2) 上記で作られたソースファイル 9 1 に対して従来のテキストだけを配置するような組み版システムに対して、画像+イメージ情報を含めて組み版できるように拡張した。

(3) 組み版された中間ファイル 9 5 からカラー画像ファイル 9 3 やカラー図形ファイル 9 2 のオリジナル情報を参照して、マルチウィンドウ 6 上にフルカラーのテキスト+画像+イメージをそのシステムの能力に合わせて表示する処理 9 7 を作成した。

(4) 上記と同様で、フルカラーで記録装置 7 に記録できるようにした。

上述した(1)~(4)の各部の詳細を順に説明する。尚、本実施例では基本的に T E X や r o f f や s c r i b e 等のバッチ型文書処理システムや W Y S I W Y G 型の文書処理システムの両方

1 1

組み版時に割り付けられる仕組となる。

例(C G M の時) :

```
\cgm {file name = example.cgm,height = 5cm,width = 4cm }
```

ここで、\cgm は C G M のファイルを include するというコマンドを示すが、コマンドは“\” (バックスラッシュ) で始まり、文書中のテキストと区別する仕組となつている。{...} の中はパラメータを示し、“file name =” で実際に include する C G M のファイル名称を示す。また、“height =”, “width =” で C G M が展開されるべき領域の高さ、幅をそれぞれ示す。

次にテキストのカラー化に関して述べるが、組み版ソースファイルに必要な拡張として、従来の白黒情報のみならず、フォントの色を明示的にコマンドで指定する方法が考えられる。すなわち、従来のフォント種、大きさに色情報とるわけであり、これはデフォルトの色を三原色(R, G, B)の割合で指定したデータベースファイル 9 8

1. 3

に適用可能であるが、説明の簡略化のためにここでは T E X 等のバッチ型システムを例にして説明する。

1) . カラー画像・図形の取り込み手段+テキストのカラー化について。

基本的に本システムではカラー画像やカラー図形ファイルは本体のテキスト組み版ファイルとは別ファイルとし、D T P の分野で標準的に利用されているフォーマットに従うものとする。

図形は C G M (Computer Graphics Metafile) という I S O 標準や P o s t S c r i p t (Adobe 社の登録商標)、画像は T I F F (Tag Image File Format = Aldus 社の登録商標) の標準フォーマットを仮定している。こうする事により、より多くのシステムで本方式の利用が可能となる。

実際に、図形やイメージを文書中に挿入するためには、以下のようなコマンドをソースファイル 9 1 に入れ、現在点 (Current Active Position = C A P) を基準として、図形やイメージ領域が

1 2

を作成し、それらの色を明示的にソースファイル中で指定する事により実現出来る。

色データベースの構築例を示すが、左端に色を示し、次には R, G, B の各色成分を 8 ビット (0 ~ 2 5 5) で示し、0 が最小の明るさ、2 5 5 が最大の明るさとして指定する。色データベースの例を第 4 図に示す。

カラーフォントの指定例として :

```
\blue (これは青色のテキスト)
```

```
\red (これは赤色のテキスト)
```

```
\green (これは緑色のテキスト)
```

のように指定すると現在の色で対応する部分を描く。

2) . カラー画像、図形、テキストの組み版手段について。

説明を簡単にするためにテキストだけの従来の組版方式について説明した後、画像や図形も加える方式について言及する。

組み版は基本的に 2 つの概念を用いて、実際のページ整形を行つている。1 つは B o x (箱) と

1 4

いう概念であり、文字フォントのようにページのどの部分にあつてもその高さ、深さ、幅が変化しないものである。

2つ目は *g l u e* (糊しろ) という概念であり、行末合せ、ページ高さ合せ等を行うために、*B o x* 間の伸び縮みしうるスペース量である。

この2つの概念を用いて、まず *B o x* 及び *g l u e* を組み合わせて文字を行方向に最適位置となるように配置を決定する。

次に、この行方向に最適配置が行なわれた文字列をページ全体にうまくフィットするように文字間のバランスを調整する仕組みとなる(第5図参照)。

また、図形や画像を含めて文書の組み版を行うには、図形や画像を、前述した幅(*w i d t h*)、高さ(*h e i g h t*)、深さ=0を持つ1つの文字フォントとして取り扱う事により容易に拡張可能となる。

これで、テキスト+図形+画像を含む文書を組み版した結果は、装置独立な *D V I* ファイル34

1 5

組み版位置を計算する際の基本単位は例えば 2^{-16} p t (p t は $1/72$ inch) 等とし、整数単位で計算しても誤差がほとんど出ない単位として計算する。

3) . カラーテキスト、図形、イメージのウィンドウ上でのプレビュー(表示)について。

このプレビュー(表示)は最近、Xウィンドウに代表されるマルチフォント、図形、イメージのカラー表示機能を持つウィンドウシステムの出現により、組み版結果を記録装置7によつてプリントアウトする前に画面上に高速に、かつ安価に結果を確認出来る事の特徴としている。

以下にそのアルゴリズムを第7図を用いて説明する。

まず、ステップS201で組み版結果を表示するウィンドウの表示枠全体のサイズを決定する。通常、このサイズは表示し得る限りの最大サイズとなるよう設定するとよい。次にステップS202で、実際に表示するウィンドウがフルカラー(RGB各8ビット)サポートか又は疑似カラー(

1 7

に格納される。*D V I* の名称は組み版結果を出力装置の解像度やフォントに依存しない形態で保持し、その結果、どのマシンで実行しても同じ結果が保証される事に由来する。この中間結果ファイルの例を第6図に示す。

図中、180は*D V I* ファイルである事を示す識別子であり、181は組み版された各ページの情報で、例えばそのページ内に含まれるフォント名称、サイズ、色及び実際のページの組み版結果情報182へのポインタから構成され、ページ数だけこのヘッダが存在する。ページ情報部182には、実際の組み版された情報を含み、基本的に以下のような情報が格納される。

- ・ フォント切換 (タイプフェイス+サイズ+色) 命令
- ・ 印字文字コード
- ・ 水平・垂直方向現在位置移動量
- ・ 矩形描画 (現在位置に指定された幅、高さ)
- ・ カラー画像情報 (ファイル名称、幅、高さ)
- ・ カラー図形情報 (ファイル名称、幅、高さ)

1 6

例えばRGBがそれぞれ3, 3, 2ビット又はRGBそれぞれ5レベルetc)かを判定し、フルカラーサポートでない時は1バイトの情報でR, G, Bがどのようにウィンドウ上で発色するかを示すLUT(Look Up Table)を設定する(ステップS203)。こうする事によつて同時に発色できる色数には限りがあるが、カラー表示のための初期化が終了する。

実際の印刷イメージとCRTの表示画面とでは解像度が異なるため、表示用の文字フォントはプリント用に使われるフォントを縮小する(ドット構成数を少なくする)必要があるし、カラー画像、図形も同様に縮小する必要がある。この縮小倍率をステップS204で決定するが、それは表示画面の縦横方向のドットサイズと利用するフォントが作成された解像度でのオリジナル文書の縦横方向のドットサイズの比較となる。

これらの準備が終了した後、文書フォーマッタ11で作成した、表示したい*D V I* ファイルをオープンした後、対応ページ情報のありかを第6

1 8

図のページヘッダ 181 をたぐる事によつて、ファイル内をシークする（ステップ S 205）。後はページ内の組版情報（182）を 1 コマンドずつスキップして（ステップ S 206）、注目ページに対する処理が終了したと判断するまで（ステップ S 207）、対応した処理ルーチン（ステップ S 209～S 219）を実行する。

これらの処理ルーチンを大別すると、以下の 5 つに分類できる。

a. 文字コマンド（ステップ S 209、S 210）

文字コマンドは文字コードをパラメータとして持ち、現在のフォント種・サイズ・色で指定された文字コードを現在位置（CAP）に表示する。フォントはフォント切替命令で指定されたものを利用し、切替時に 1 バイト（欧文）、2 バイト（和文）フォントデータ 32 のアクセス手法に関する処理手順へのポインタが得られる仕組みである。もし、ウィンドウシステムがフルカラーサポートの場合はフォント色として、色データペー

スで指定されている R、G、B は 8 ビット×3 情報を指定するし、そうでない時には 2.4 ビットから 8 ビットへ R、G、B おのおの変換した結果でフォント色属性を指定する。その結果本ステップではポインタで示される関数に文字コード及び現在位置情報を与える事によつて、ウィンドウ上に色文字が表示される。ただ、ウィンドウシステムの形態やフォントアクセスに関する差異により文字表現の方法は一様ではない。

例えば、X ウィンドウのようなサーバ・クライアントモデルでも表示するためのフォントがクライアント側に依存する場合とサーバ側に依存する場合利用方法が異なる。

大きな相違点は、クライアント側がフォントをアクセスするか、サーバ側がフォントをアクセスするかである。クライアント側がフォントをアクセスする時には、ステップ S 204 で決定された縮小率に従つてフォントを縮小し、ウィンドウに表示するのに最適なフォントサイズに変換した後、フォントイメージをサーバに送出する。

19

それとは逆にサーバ側にフォントが存在する場合には、フォントの拡大、縮小はクライアントから容易には実行できないため、サーバ内に存在するフォントのうち最も近いフォントを選択して、文字コードのみクライアント側から送出する。

b. 画像コマンド（ステップ S 211、S 212）

画像を文書中に貼り込む際にまず、オリジナル画像を画像表示サイズ（先に説明された width、height）へ変換する事とそこからウィンドウに表示するための倍率に変換する。この 2 つの変換を組み合せ 1 つの変換で表現すると：

orig-width : オリジナル画像の幅方向有効ドット数（オリジナル解像度）

orig-height : オリジナル画像の高さ方向有効ドット数（オリジナル解像度）

dest-width : 組み版する幅方向のドット数（ステップ S 204 で仮定した解像度での）

dest-height : 組み版する高さ方向のドット数

20

（ステップ S 204 で仮定した解像度での）

$$\text{変換倍率} = \min \left(\frac{\text{dest-width}}{\text{orig-width}}, \frac{\text{dest-height}}{\text{orig-height}} \right)$$

×ステップ S 204 の倍率

で求まる。これはどういう事を示しているかというと、オリジナルの画像が 400 DPI で 100 × 100 dot のサイズであるとする、それを（1 cm × 2 cm）の幅、高さの領域に印字したいとする。そしてその結果を 100 DPI のウィンドウ上に表示したいと仮定すると、変換倍率は、

$$\min \left(\frac{2/2.54}{1/4}, \frac{1/2.54}{1/4} \right) \times \frac{1}{4} = \frac{1}{2.54}$$

の倍率となる。

ここでは、x、y 方向に同じ比率での拡大、縮小を仮定している。この変換倍率に従つて、オリジナル画像自体をまず拡大、縮小するが、縮小時には単純に x、y 両方向の間引きで行ない、拡大時には x、y 両方向への水増しすることで行う。これ以外にも、近傍画素で論理和をとりながらの

21

22

拡大etcのようにいろんな方法も選択出来る。このように表示用に画像サイズを変換後には、次の処理としてオリジナル画像の色精度と表示側の色精度とのギャップをなくすべく、色の深さ方向の変換処理が必要である。

ここで、色精度として代表的な3種類、白黒(1ビット)、疑似カラー(8ビット)、フルカラー(24ビット)を考える。そうすると、第8図に示すような変換パターンが考えられる。

ウィンドウの色精度に対応した表示のための画像が得られると、実際にその情報+現在位置をサーバ14に送出して、ウィンドウ上にカラー画像を表示する。

C. 図形コマンド(ステップ213, 214)

図形の場合も画像と同様にオリジナルの図形を組み版サイズに拡大、縮小する倍率にステップ204で求めた拡大、縮小倍率をかけて、最終的にオリジナル図形座標系からウィンドウ座標系への変換式をまず求める。

その後、図形ファイルをスキャンして図形要素

2 3

フォントに切り換える命令をクライアントからサーバに送り、サーバ内部でフォント切換にともなう処理を実行する。

4). カラーテキスト、図形、イメージの印刷について、

ここでは印刷機能は、カラーDTPにおいて必要なカラー印刷装置、例えばカラーLBPやカラーインクジェット装置をホスト側から組み版結果に従い制御する仕組みについて説明する。ここで記録に関しては2種類の方式に分けて説明する。1つはカラーPosc Scieipプリンタに見られるようにプリンタ側にページメモリやコマンドインタプリタを装備し、ホスト側からの指令に従い印字する装置である。もう一方は、ホスト側にプリンタの色精度に対応する深さ方向分のページメモリを持ち、通信回線あるいはバスなどを介して情報を記録装置に送って印刷する方式である。この両者を比較すると、ホストマシン側の負荷は断然後者の方が重く技術的課題が多い。

以下に、各々の方式について説明する。

2 5

を見つけると、その座標に対して上記の変換式を施して、ウィンドウ上での座標に変換し、該当するウィンドウに対する図形描画ルーチンをコールする形式となる。

ここで、図形の色要素が変換されるたびにテキスト部で述べたように色指定コマンドを送出して、現在の色属性をupdateしておく。

d. 位置移動コマンド(ステップ215, 216)

現在位置(CAP)座標を内部的に保持している変数h, wに縦又は横方向の移動量に応じた量を加算して、現在位置(x, y)を保持する。

e. フォント切換コマンド(ステップ217, 218)

印字するためのフォント(タイプフェイス、サイズ情報、色etc)を切り換える命令が入力されると、例えばXウィンドウのクライアント側にフォントがある形態だと、クライアント内のフォントアクセスのための関数へのポインタを切り換える。逆にサーバ側にフォントがある場合には最適

2 4

4-a. 記録装置側にインタプリタを持ち記録イメージを生成する場合。

基本的な処理の流れは、3). の表示で示した第7図のフローと似ているが、組み版結果から記録装置のページ記述言語に変換する際の留意点を述べる。

a). 画像やイメージを決められたサイズに変換するためのスケーリング。

もし、ページ記述言語自体にスケーリングをサポートする機能があればそれを利用する。もし、なければ3). で述べたようにオリジナルの画像はホスト側で拡大、縮小率に従い変換し、図形の位置情報はスケーリングを加えて記録装置にコマンドを送出すればよい。

b). 記録装置の色情報に対応した色変換する。

記録装置の持つ色精度に従いオリジナルのテキスト、画像、図形の色コマンドを変換して記録装置に情報を送出する事で記録が出来る。

4-b. ホスト側にページメモリを持ち、情報を展開

2 6

した後、記録装置にページメモリ情報全体を送出する場合。

本方式では、記録装置側にページ記述言語（例えばPost Script）を解析して記録する機構がない場合を仮定し、ホスト（本装置）に設けられたページメモリ22を用い、そのページメモリ上にテキスト、図形、画像情報を展開した後、ビデオ信号として記録装置に情報を転送して印字する。

処理のキーとなるのは出力装置の色精度、すなわち白黒（1ビット）、疑似（8ビット）、フルカラー（24ビット）にあわせて動的にページメモリ22に深さを考慮してメモリを割り当て、そこにオリジナル情報の色情報度を記録装置側の色精度に合致するように変換処理する事である。

このページメモリ配置の例を第9図を利用して説明する。

・まず、白黒のビットマップに関しては、同図（A）に示すように、行方向に走査して8ビットを1バイトにパックして情報を格納する。又ドットへのアクセスが容易となるように1行nドットの

27

り当てる。そして、記録装置に送る場合は、

イ）R、G、Bのプレーン毎に送出する。

ロ）R、G、Bの順に1画素ずつ送出する。

の二つの方法が考えられるが、これは記録装置に依存するので、最初に記録装置を確認して対応すればよい。

ここで、注意すべき点は、白黒ビットマップ例では“OR”論理、すなわち、すでに描画されているイメージに論理和をとる事は意味をなすが、フルカラーや疑似カラーにおいては論理和をとると予期した色を出す事は不可能であり、指定した色でのテキスト、画像、図形情報をページメモリ上に“SET”する論理を採用する。この処理は基本的に後から描いた情報が優先される方式であり、カラー情報の表現能力に問題がある。

そこで、表現能力を高めるために、オーバーレイプレーンやアルファプレーンの導入する手法が考えられる。

オーバーレイプレーン

これは通常のカラーテキスト、画像、図形を

29

時に、行の終端がバイト領域に合致するように、 $8 - (n \bmod 8)$ ビット分パディングを行う。そして、次の行の情報が次のバイトより連続して割り当てられる。尚、ここで使用した“mod”であるが、“ $A \bmod B$ ”とは、AをBで割ったとこの余り返す関数である。

・疑似カラーでは前者のビットマップが、ここでは同図（B）に示すように、8枚重なったものと考え、1画素単位に1バイト（8ビット）で表現される仕組みとなる。ここで、一般的な1バイト中でのカラー情報の表現方法として、3ビットをR及びGに、2ビットをBに割り当てる。

又、記録装置によつては、色を持たないが白から黒への濃度階調を持つ事により濃淡再現出来るものもあるが、その際には8ビットで計256階調表現出来る。

・フルカラービットマップでは同図（C）に示すように、R、G、B毎に3つの独立したページメモリを割り当て、R、G、Bそれぞれ1画素に対して1バイトずつ専有し、行方向順にメモリに割

28

ページメモリのフルカラー又は疑似カラープレーンに展開するとともに、例えば、白黒のテキスト、画像、図形etcを1ビット深さのオーバーレイプレーンに展開し、オーバーレイプレーンの白黒情報をカラープリンタに優先させて印刷する（第9図（D）参照）。

別の問題点として、A4サイズに400DPIの解像度で出力する際には、白黒で2Mバイト、疑似カラーで16Mバイト、フルカラーで48Mバイトのページメモリの容量が要求される。

白黒の場合は別として、疑似カラーやフルカラーのページメモリがWS（ワークステーション）で確保出来る形態であれば問題ないが、通常のWSでは一度に確保する事は困難な場合が多い。そこで、解決策として、ページメモリの確保及び描画を何回かに分割して、それらの描画イメージを連結する事によつて最終了描画イメージを作成する。第10図を参考にしながら説明すると、まず、疑似カラーメモリの分割個数をシステムの1度に割り当可能容量と比較して決める。も

30

し、システムのその容量が5Mバイト程度だとすると、 $16M/4=4M$ バイトで疑似カラーメモリを列方向に4分割する(ステップ300)。そして、DVIファイルの第1スキヤンで列の数全体を m とすると、 $0\sim m/4$ ライン分の疑似カラーページメモリを確保し、関連するテキスト、イメージ、図形情報をこの疑似カラーページメモリに書き込み(ステップ304)、DVIファイルの読み込みを終了した後、疑似カラーページメモリの情報をファイルに書き込む(ステップ305)。

そして、分割数分のページメモリへの書き込みが終了すると、それらの個数分のファイルを連結して1つの完全な疑似カラーページメモリを作成する(ステップ307)。

ここでは、疑似カラー例を示したが、フルカラーに関しても同様にして実現出来る。

最後になるが、テキスト、図形、イメージのページメモリ上への描画のアルゴリズムは第7図のプレビューの概略フローと同様であるが、異なる。

3 1

[発明の効果]

以上説明したように本発明によれば、カラーテキスト、カラー画像更にはカラーイメージを編集することができると共に、それらの編集結果を接続される表示装置や印刷装置の機能に合わせて出力することが可能になる。

4. 図面の簡単な説明

第1図は実施例における文書処理システムのブロック構成図、

第2図は従来のバッチシステムの処理の流れを示す図、

第3図は実施例におけるメインフローを示す図、

第4図は色データベースの例を示す図、

第5図は文字間のバランスを調整する原理を示す図、

第6図は組み版結果ファイルのデータ構造を示す図、

第7図は文書表示に係るフローチャート、

第8図はオリジナル画像と表示装置の相違に基

るのはステップ204の倍率決定は表示イメージがプリントイメージに対して縮小されるため必要であるが、プリント時にはこのステップは省略する。

以上説明したように本実施例によれば、従来のバッチ型システムで困難であったテキストと図形・画像をカラー化して統合的に取り扱う環境が実現可能となる。

又、本実施例で説明したように種々のタイプの記録装置やウィンドウシステムに対応出来るようにシステムが装置独立性を考慮して構成されているため、広範なシステム、又他システムとのデータ変換において効果を発揮する。

尚、本実施例ではバッチ型の文書処理システムを基本に説明したが、WYSIWYG型の文書処理システムにおいても、ユーザインターフェース部を除けば基本的に組み版に必要な情報、組み版結果、表示、印刷というモジュールで構成されるため、本実施例の方式を適用する事が可能である。

3 2

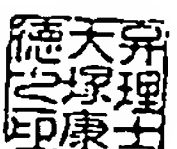
づく変換内容を示す図、

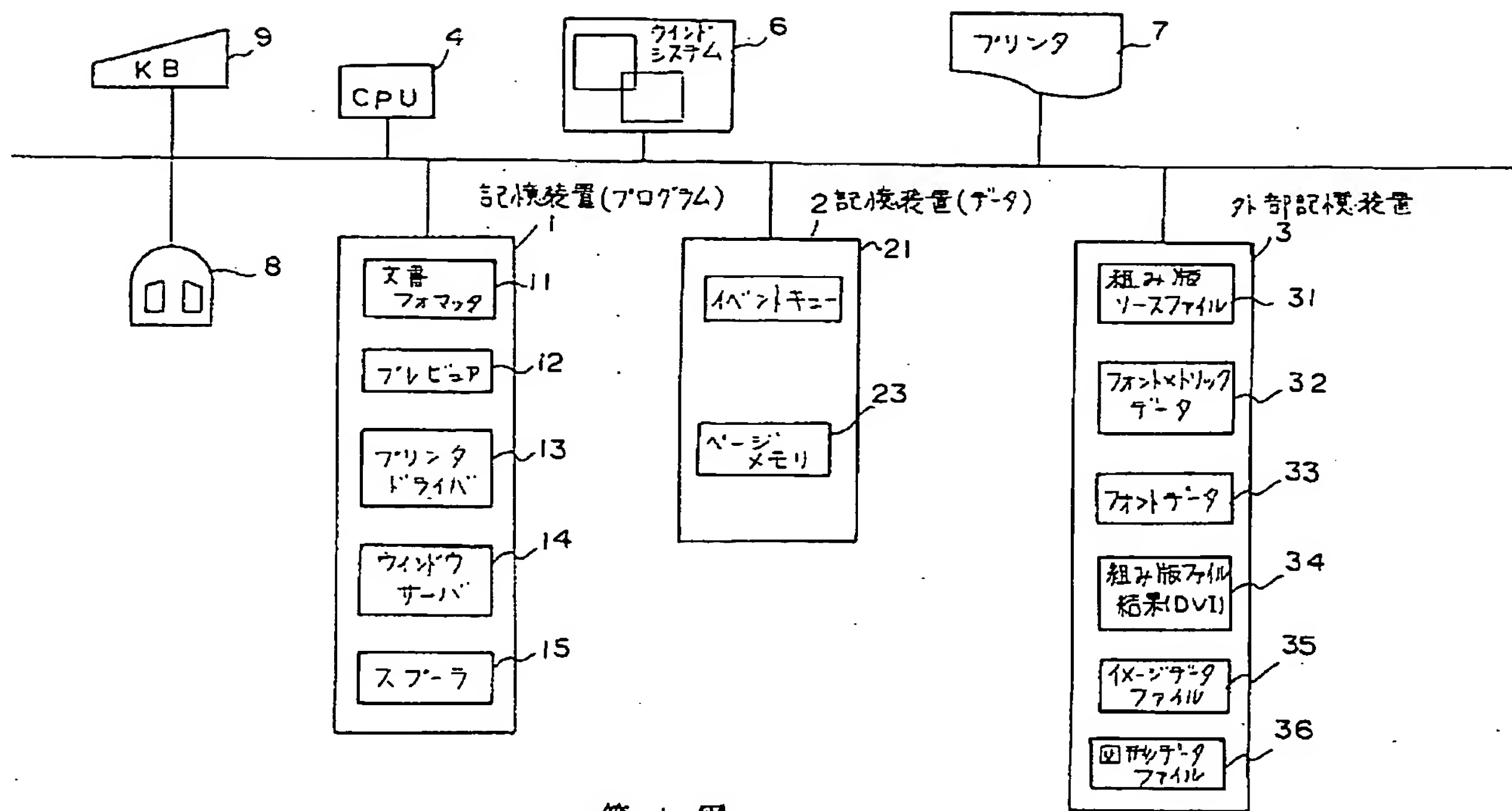
第9図(A)～(D)は第8図に示した変化内容に基づく表示用メモリの活用の仕方を示す図、

第10図は印刷出力時の印刷ファイルの作成手順を示すフローチャートである。

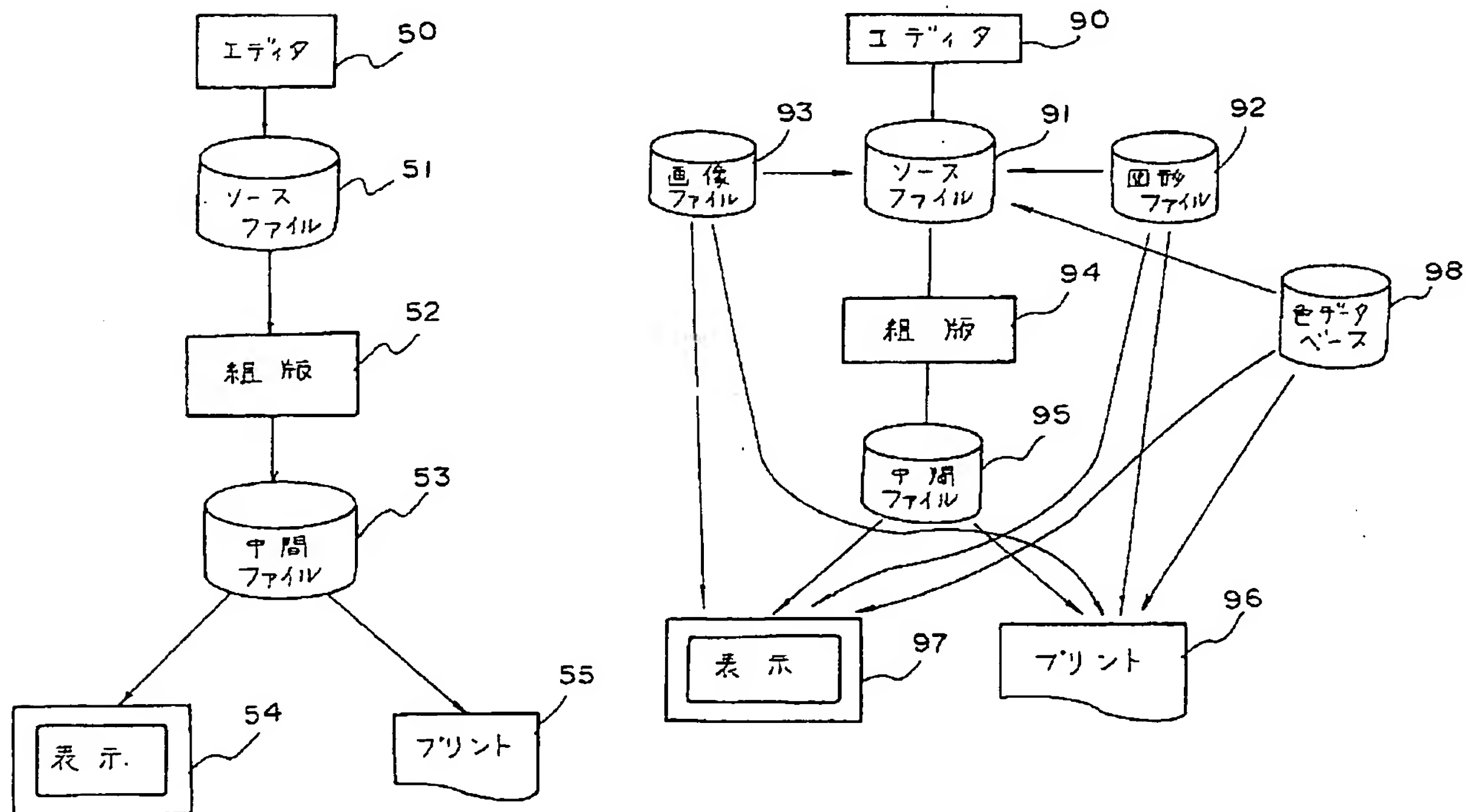
図中、1及び2…記憶装置、3…外部記憶装置、4…CPU、6…ウィンドウシステム、7…プリンタ、8…マウス、9…キーボード、11…文書フォーマッタ、12…プレビュー、13…プリンタドライバ、14…ウィンドウサーバ、15…スプーラ、21…イベントキュー、22…ページメモリ、31…組版ソースファイル、32…フォントメトリックデータ、33…フォントデータ、34…組版ファイル結果(DVI)、35…イメージファイル、36…図形ファイルである。

特許出願人 キヤノン株式会社
代理人 弁理士 大塚康徳(他1名)





第 1 図

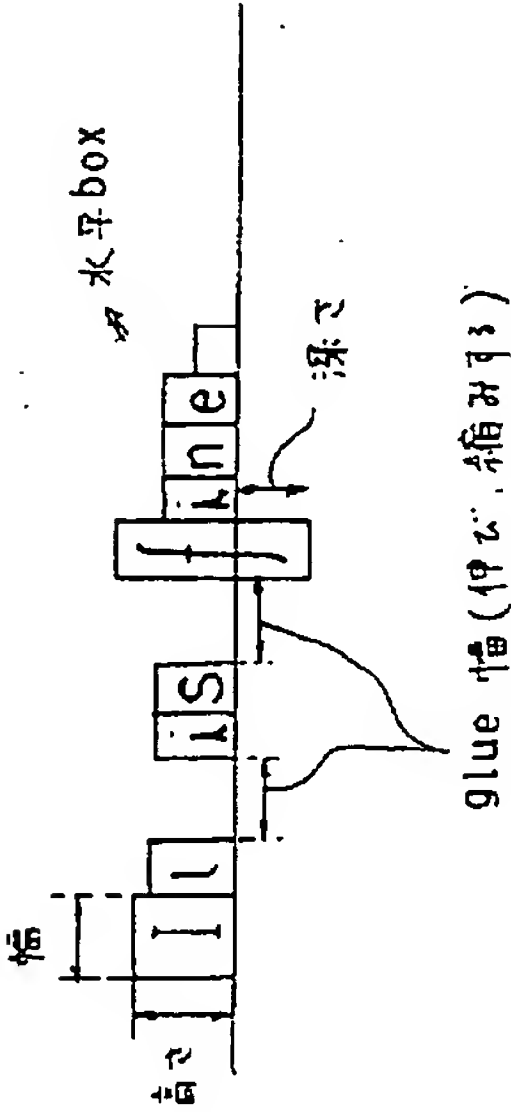


第 3 図

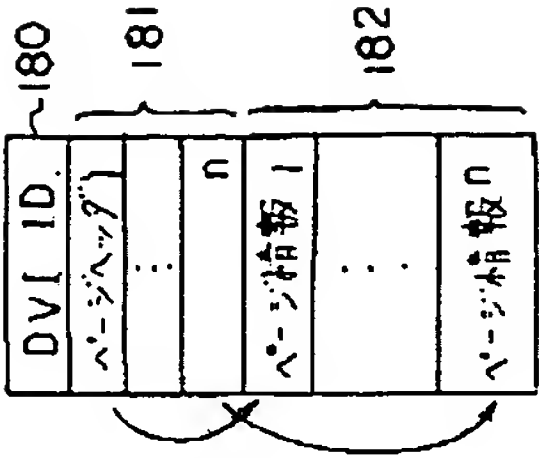
第 2 図

色	R	G	B
white	255	255	255
black	0	0	0
green	0	255	0
red	255	0	0
blue	0	0	255
yellow	255	255	0
cyan	0	255	255
magenta	255	0	255

第 4 図



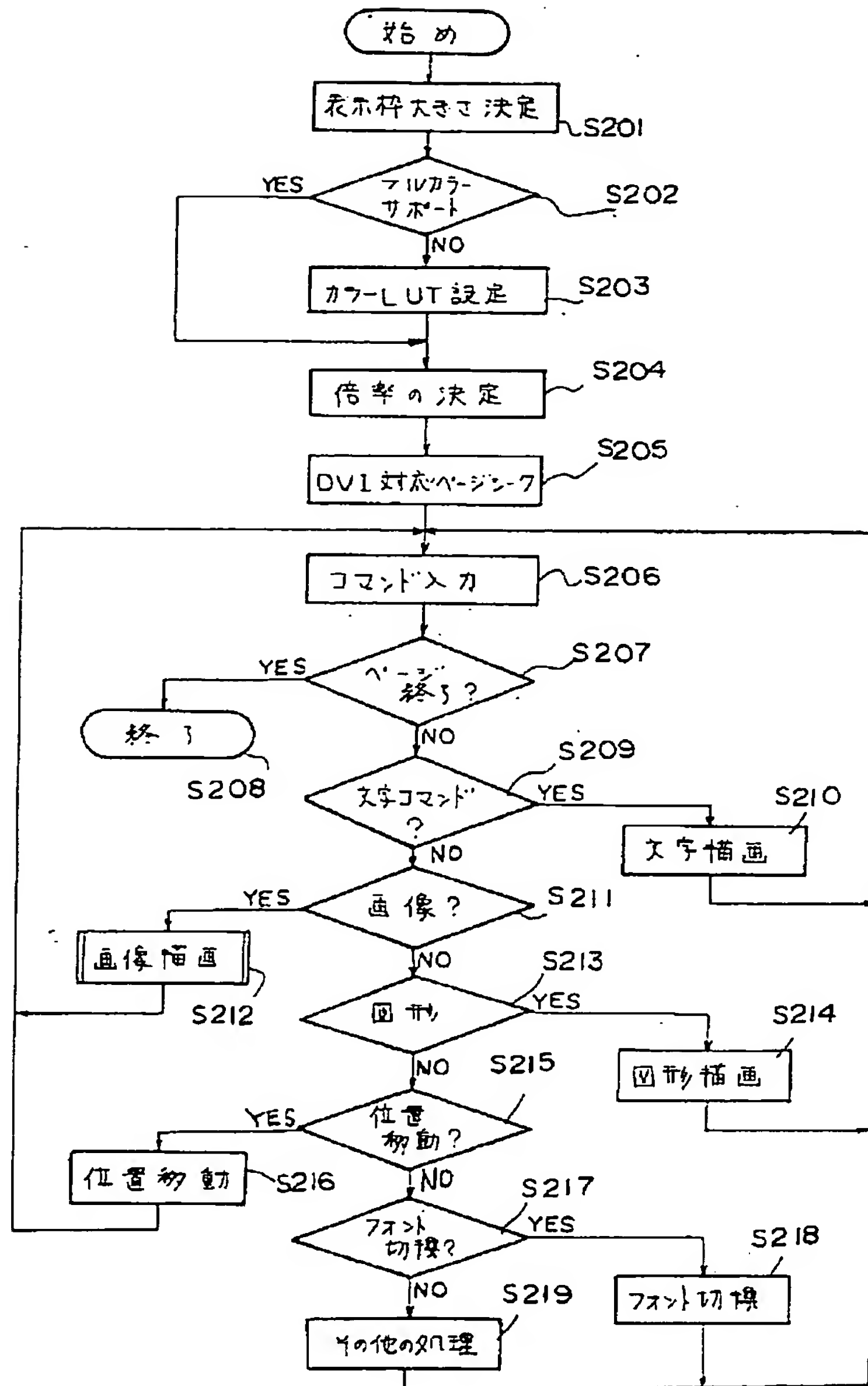
第 5 図



第 6 図

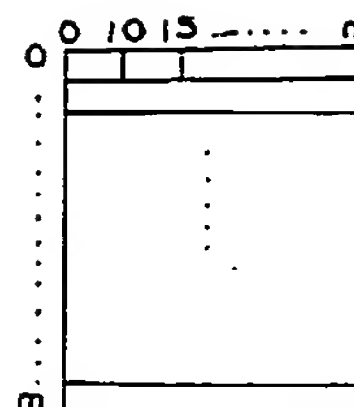
画像表示	白黒 (1bit)	疑似色 (8bit)	フルカラー (5bit)
白黒 (1bit)	そのまま 白黒	あり色で"ダイザ" 1bitに	あり色で"ダイザ" 1bit
疑似色 (8bit)	8bit拡張 白黒	LUT変換 (疑似カラー-間)	"ダイザ" 誤差拡散 単純単色
フルカラー (24bit)	24bit拡張 白黒	疑似カラー-間 フルカラー-変換	そのまま

第 8 図

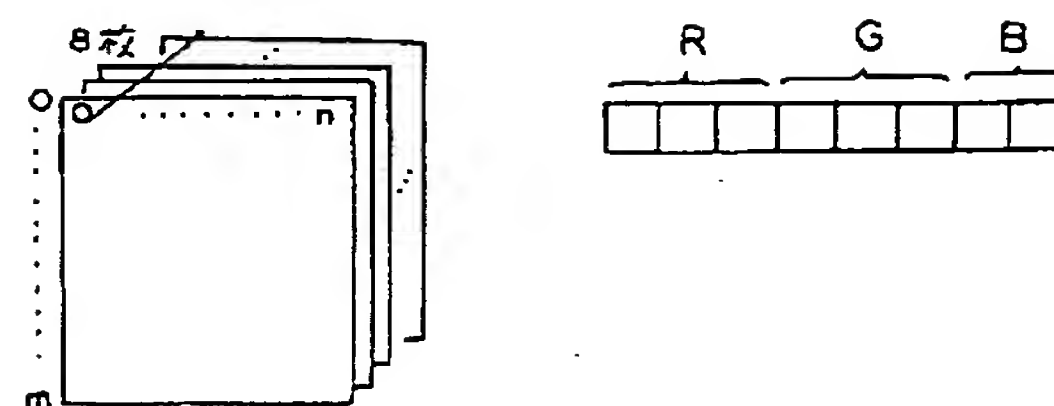


第 7 図

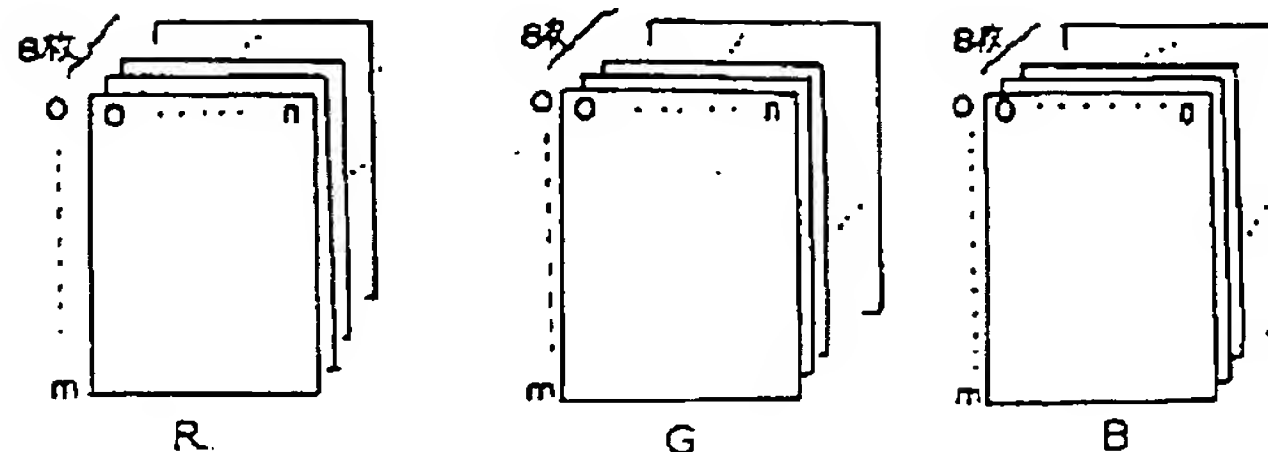
(A) 白黒(ビットマップ)



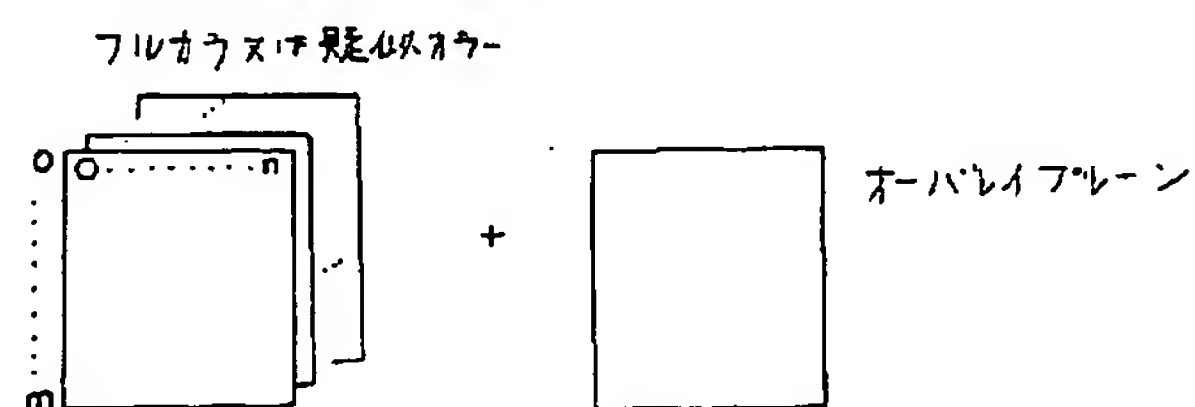
(B) 疑似カラー



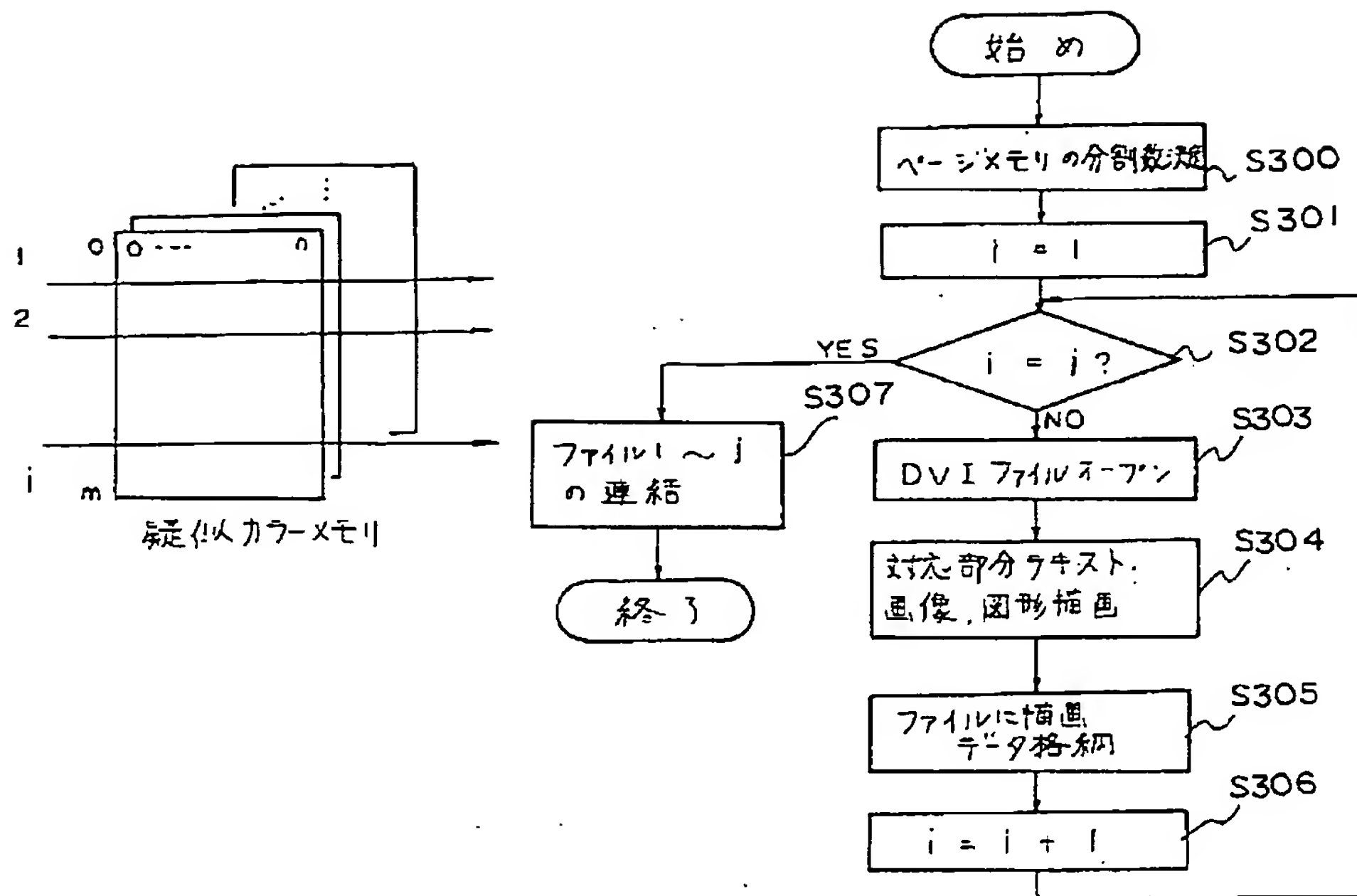
(C) フルカラー



(D) 疑似カラー又はフルカラー+テキストプレーン



第 9 図



第 10 図